

AMENDMENTS TO THE SPECIFICATION

Please amend the specification by replacement paragraph as follows:

[0029] Referring now to Fig. 3, a first tier logic block is shown. First tier processing [300] involves interaction with application programs and provides an LUN/LBA storage interface. In this first tier LUN/LBA processing block [300], an application program data 302 provides reference data to be hashed, and for which an OID is generated. A comparison is made between a reference data OID value and OIDs stored in a high-level OID table 304. From the comparison, a determination is made as to whether reference data for which an OID has been generated, has been previously written. The present invention allows for three different types of LUN/LBA access. The access type may be configured by a user (e.g. system administrator) and is determined subsequent to a previously mentioned determination step.

[0031] If a write-many property is enabled 314, high-level OID table 200 is consulted to determine whether a particular LUN/LBA 202 is associated with an OID 204. If LUN/LBA has a null OID 316, write operation to LUN/LBA 332 is allowed. Subsequent to write operation to LUN/LBA 332, OID is written to high-level table 334. If LUN/LBA has an OID 318 [310], then application program data 302 is allowed a re-write operation to LUN/LBA 320 and a previous OID is over-written in a high-level OID table 322.

[0034] Referring now to Fig. 4, a second tier logic block is shown. Second tier OID processing [400] receives content reference data from a first tier interface 402 or directly from an application 404. Thus, an application can directly opt to use the OID interface of the second tier [400] instead of accessing data via the LUN/LBA interface of the first tier [300] of Fig. 3. In second

tier OID processing [400], content reference data to be written is hashed and an OID is generated 406. Next, a high-level OID table is updated 408 with newly generated hash value for an associated LUN/LBA 202. Low-level OID table is further updated 410 with newly generated hash value. Second tier OID processing [400] helps to eliminate duplicates - if two separate LUN/LBA combinations have the same data content, then content associated with both combinations will hash to the same OID. LUN/LBA counter is incremented 412.

[0035] Referring now to Fig. 5, a third tier logic block is shown. Third tier OID processing [500] receives data from second tier interface 502. Low-level OID table is accessed 504 to write to an LUN/LBA 506 data received from second tier interface 502. If data from a particular LUN/LBA combination is to be retrieved 508, reference data it is re-hashed and the hash value generated is compared against an OID value 510 generated during the original writing of data in the second tier logic block [400] of Fig. 4. In this tier, data is simply written or retrieved from a lower level LUN/LBA of a storage subsystem. Thus, third tier OID processing [500] helps to verify the accuracy of lower level LBA content that has been retrieved 508 from a storage subsystem.